2009 11th IEEE International Symposium on Multimedia

# Bowed String Sequence Estimation of a Violin Based on Adaptive Audio Signal Classification and Context-Dependent Error Correction

Akira Maezawa, Katsutoshi Itoyama, Toru Takahashi, Tetsuya Ogata, Hiroshi G. Okuno
Department of Intelligence Science and Technology
Kyoto University, Japan
Email: {amaezaw1,itoyama,tall,ogata,okuno}@kuis.kyoto-u.ac.jp

***Abstract*—The sequence of strings played on a bowed string instrument is essential to understanding of the fingering. Thus, its estimation is required for machine understanding of violin playing. Audio-based identification is the only viable way to realize this goal for existing music recordings. A naïve implementation using audio classification alone, however, is inaccurate and is not robust against variations in string or instruments.**

**We develop a bowed string sequence estimation method by combining audio-based bowed string classification and context-dependent error correction. The robustness against different setups of instruments improves by normalizing the F0-dependent features using the average feature of a recording.**

**The performance of error correction is evaluated using an electric violin with two different brands of strings and and an acoustic violin. By incorporating mean normalization, the recognition error of recognition accuracy due to changing the string alleviates by 8 points, and that due to change of instrument by 12 points. Error correction decreases the error due to change of string by 8 points and that due to different instrument by 9 points.**

## I. Introduction

There is a growing demand for machine assessment of *fingering* techniques of the violin and other stringed instruments. *Fingering* refers to a sequence of fingers pressed and strings bowed required to play a music score. It is particularly relevant in violin performance because it influences playing facility and produced tone.

A note may be played on different strings, using different fingers. Typically, different choice of the pressed finger changes the playing difficulty, and that of the bowed string changes the timbre. For example, violinists may play a piece with as little effort as possible by avoiding fingerings that excessively use the little finger (typically the weakest finger). On the other hand, a violinist may choose to play a piece in a particular bowed string to generate the "right" sonority, though it may not be the easiest fingering. Whether or not a particular bowed sequence sounds "right" is subjective, and is a matter of personal taste. Needless to say, violinist cannot play in a fingering that sounds "right" if it is physically too difficult for him/her to manage. Therefore, the quest for the "right" sound is constrained by the violinist's technical limitations.

Most existing works in fingering assessment focused on finding the "best" way to finger a piece. For example, Lu et al. [1] extracted, through audio-visual fusion, the finger motion, along with other aspects of violin playing for automated violin tutoring. Other works have focused on finding the "optimal" fingering [2] to play a given passage, i.e. one that minimizes some kind of physiological cost functions. Little attention was paid to the control over different timbre a violinist has in his/her arsenal by choosing different bowed string sequences.

We focus on another application of fingering assessment – extraction of a sensible fingering required to re-create a particular audio recording. As discussed later, fingering is influenced by both personal taste for a particular tone and technical limitation. Therefore, our application is important for uncovering personal taste of the violinist that recorded the audio. For example, it could help an intermediate-level violinist extract, from a CD recording of an avant-garde artist, a fingering that allows him/her to sound more like the avant-garde. It may also help music historians easily gather fingering data to analyze stylistic differences in violin fingering[1].

Our application requires the estimation of bowed strings using only an audio signal and a music score, instead of other methods such as fingering extraction from audio-visual fusion, or optimal fingering estimation from the music score. Audio-visual fusion [1], [3] is inviable because a video rarely shows the required information to extract fingering throughout the entire performance. For example, video recording of a violin concerto would shoot not only the violinist's fingers, but also the facial expression or the orchestra as well. Moreover, there are substantially fewer video recordings of a violin performance compared to audio recordings. Therefore, audio-visual fusion supports only a small subset of existing recordings. Optimal fingering estimation using score-based constraints [2], [4]–[6] fails because it is inherently incapable of listening for different sonority created by the different choice of strings bowed.

We estimate the bowed strings (and not the pressed

[1]Casual discussions with a violinist of the "Flesch" school and violinists from the "Galamian" school suggested that different fingering preferences do exist.

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.30

fingers) because only bowed string is the factor pedagogues discuss on fingering that pertains to differences in the sonority [7]–[9]. In other words, the information necessary and sufficient to aurally re-create a given recording is the sequence of bowed strings[2]. The necessary sequence of finger placement may be complemented by, for example, using the optimal fingering algorithm constrained to play on a specific bowed string sequence.

What is so hard about estimating the sequence of bowed strings? Bowed string estimation is difficult because there are many elements other than the bowed strings that significantly modify the timbre of the violin. Namely, playing a passage on a different violin, different brand of strings, or with different bow stroke all create audible differences comparable to playing on a different string. Fortunately, the effect of different bow strokes may be nullified by preparing the training data with a wide variety of bow strokes. However, it is unrealistic to nullify the timbral differences due to using different strings or violin – recording the training data on all violin that exists to date, on all combination of strings is impossible. Therefore, robustness against variation of instruments and strings is required for our application. Existing string estimation method based on classification using Multiple Discriminant Analysis of the principal axes of the power spectra [10] offers no quantitative results, but our implementation suggests that it is not robust against such variations.

In this paper, we present a new bowed string sequence estimation method that is robust against variations of brand of strings and instruments. Our method is based on integrating audio-based classification and error correction based on analyzing the musical context (the music score). We also adapt the trained model to a violin with different acoustic properties, caused either by changing the strings to a new brand, or by playing on an entirely different instrument.

We incorporate error correction to ameliorate the low recognition accuracy attained when using the audio alone. Violinists would often listen for neighboring notes or abrupt changes in sound to infer the string on which a note was played. In fact, many recognition errors generated are "obvious" to violinists with sufficient experiences because the errors are unrealistically complicated and unmusical. We attempt to correct errors based on such heuristics.

Section II summarizes the fingering techniques, and Section III describes our system. Our error correction and audio-based classification are introduced in Section IV and V, respectively. Our mechanism for adaptation is discussed briefly in Section VI. In Section VII, we evaluate the performance of our error correction scheme.

[2]There are effects based on fluctuation of pitch that has a secondary consequence of changing the timbre, but that is more a problem of pitch trajectory than timbre.

## II. Rudiments of Violin Fingering

We shall briefly review the essences of violin fingering and its terminologies, as they play important roles in understanding our error correction algorithms.

The left hand defines the pitch and the string on which a note is played. The art of determining the sequence of finger placed on the string and the string on which a note is played is known as the "fingering [8], [9]." While there are two defining factors in fingering (finger and string), violinists often talk about fingering also in terms of the left hand position ("position"), the general placement of the left hand required to play a given note in a certain string using certain finger, as shown in Figure 1.

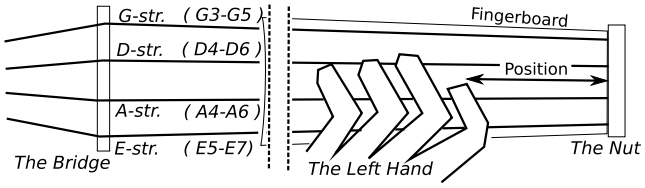


Figure 1. Left Hand Terminologies

Finger placement is determined by considering technical ease and musical effect [7]. Using certain finger (e.g. index finger instead of the little finger) facilitates execution of some musical effects related to pitch, such as smooth transition between two notes (*glissando*), or a low-frequency modulation of a note (*vibrato*). At the same time, some sequence of finger placement is easier to execute than others. For example, rapid movement of the little finger is considerably more tiresome than that of the index finger.

Bowed string is determined by considering the consistency of sonority [7]. For example, since each string has a distinct sonority, violinists often play on one string to prevent abrupt changes of the sound quality. Bowed string is the only factor in fingering that produces audible differences.

Position is determined often for visual effect and technical facility of playing. For example, violinist may present a "flashy" playing style through a wide change of position [9].

The consistency of sonority offered by playing on one string and the consistency of position which facilitates playing are often at a trade-off. In a fast piece, some abrupt changes of sonority caused by a certain fingering may be overlooked if it simplifies playing. On the other hand, in a slow piece, a violinist may choose a difficult fingering that produces a certain sonority. A violinist may, for example, value consistency of sonority in a slow "singing (*cantilena*)" passage by playing on one string [7].

## III. System Setup

Our method is a four-stage procedure, as shown in Figure 2. First, we align the score and the audio. Next, we determine

the average feature and normalize the features of the entire piece by the mean. Next, we estimate the bowed string using a trained model. Finally, we analyze the resulting sequence of bowed strings to find if there are any possible "errors."

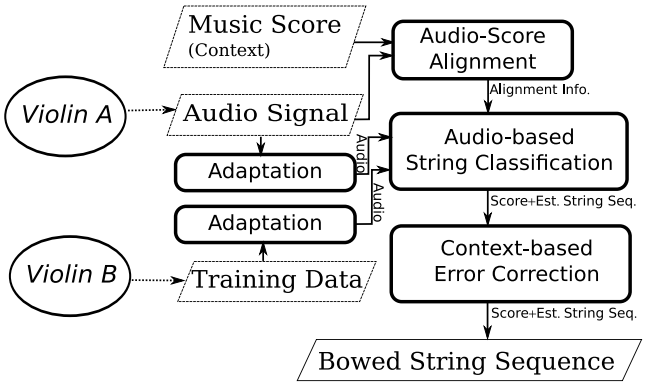


Figure 2. System Block Diagram

## IV. Context-Dependent Error Correction

We would like to incorporate musical context to detect incorrectly identified strings. By observing the bowed string usage of various violinists, we have noted the following recurrent tendencies in the choice of strings bowed:

- **Consistency of Pitch / String Transition Direction (Rule PST-D)** : Given a sequence of two notes neither of which is an open-string, if the second note is higher than the first, it uses the same or higher-tuned string. Likewise, if the second note is lower than the first, it uses the same or lower-tuned string. Figure 3 shows an example.

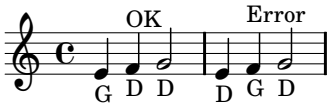


Figure 3. Example of Rule PST-D

- **Consistency of Pitch/String Transition Magnitude (Rule PST-M)** Given a sequence of three notes, consecutive jumps between nonadjacent strings occur only when intervals between the first and the second, and between second and third are greater than perfect fifths (7 semitones) apart. Figure 4 shows an example.

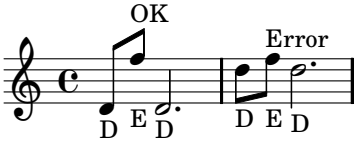


Figure 4. Example of PST-M

- **Consistency of Bowed String Usage in a Mordent (Rule MORD)** A mordent-like phrase is played on the same string (we define a mordent-like phrase as a rapid alternation between a note and a note above or below it). Figure 5 shows an example.

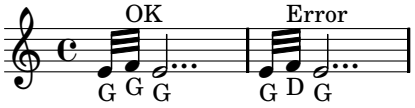


Figure 5. Example of MORD

Our observations are probably attributed to violinists' general reluctance towards complicated fingerings that serve no musical ends [8].

For example, violating PST-D requires an abrupt change of sonority, possibly accompanied by an abrupt change of the position. Violinists either focus on maintaining consistent sonority by playing on the same string (for aesthetic purposes), or crossing strings to ease technical demands. In this light, violating PST-D appeals neither to aesthetics nor technical ease.

PST-M occurs perhaps because interval of less than a perfect fifth could be played on two adjacent strings without a change of left hand position. Since consecutive jumps between nonadjacent strings are technically more demanding than those between adjacent strings, such jumps are technically sensible only when it eliminates the need for change of position. Aesthetically, consecutive jumps create abrupt changes of tone, so violating PST-M would occur if there is a reason to emphasize one particular note. That, however, is typically attained through changing the bow stroke and not through playing consecutive jumps to nonadjacent strings.

MORD occurs because a mordent tends to be played in one bow stroke. Typically, rapid alternation of bowed string within one stroke involves quick "snap" of the right wrist or the arm. Such motion is considerably more tiresome than lifting of a left finger. Therefore, MORD is violated only when there is strong aesthetic need to play a mordent by a rapid string crossing.

We define a note that disobeys the rules introduced above as an *error*, and derive algorithms to correct such errors. Let $\mathrm{S} = \{S_1 \cdots S_N\}$ be a monophonic score of length $N$ where $S_i$ contains the $i$th pitch as a MIDI note number, and let $\mathrm{St} = \{\mathrm{St}_1, \cdots, \mathrm{St}_N\}$ contain the sequence of string identified for the score, where $\mathrm{St}_i$ indicates the string identified for the $i$th note. $\mathrm{St}_i$ of 1, 2, 3 and 4 respectively indicates the G-string, D, A, and E.

Error violating PST-D is fixed by Algorithm 1, and PST-M by Algorithm 2. We apply Algorithm 1 and 2 repeatedly for a fixed number of iterations. MORD is imposed by constraining the estimated bowed string to be the same for all notes constituting a mordent. Mordents are determined

by finding a sequence of three notes such that the following holds:

1) Duration of two of the adjacent notes are shorter than a dotted 32nd note.
2) Pitches of the first note and the third note are the same.
3) The second note is no more than two semitones apart from the first.

**Algorithm 1** Error Correction Algorithm: Observation 1

**procedure** correct-PST-D( S,St )
**for all** OP $= \{\geq, \leq\}$ **do**
  $(i_b, i_e) := (1, 1)$
  **repeat**
    **while** $i_e < |\mathrm{S}| \wedge (\mathrm{S}_{i_e}$ OP $\mathrm{S}_{i_e+1})$ **do**
      $i_e := i_e + 1$
    **end while**
    **if** $i_b \neq i_e$ **then**
      St := *fix-rule-1* ( $i_b$, $i_e$, S, St, OP)
    **end if**
    $i_b := i_e + 1$
  **until** $i_b > |S|$
**end for**
**return** St
**procedure** fix-PST-D( S,St )
err:= $\{j \in [i_b + 1, i_e] | \neg(\mathrm{St}_{j-1}$ OP $\mathrm{St}_j)\}$
**for all** $i \in$ err **do**
  **for all** $(u, v) = \{(i, i-1), (i-1, i)\}$ **do**
    **if** $\neg$ ($S_u$ played on $St_u$ is an open-string) **then**
      $(\mathrm{temp}, \mathrm{St}_u) := (\mathrm{St}_u, \mathrm{St}_v)$
      $a_{new} := \{j \in [i_b + 1, i_e] | \neg(\mathrm{St}_{j-1}$ OP $\mathrm{St}_j)\}$
      **if** $|a_{new}| = |err|$ **then**
        $\mathrm{St}_u :=$ temp
      **else**
        **break**
      **end if**
    **end if**
  **end for**
**end for**
**return** St

## V. Extracting and Modeling the Violin Sound

We assume that existing methods are used to align the score and audio [11], and to separate out the violin part fairly accurately [12]. Next, we extract the output energy of a filter bank from the separated signal as the feature, because we found that it is highly discriminative for classifying violin strings.

Since playing on different environment (different brand of strings or different violin) creates an audibly significant change, we need to adapt our extracted features to a new environment.

**Algorithm 2** Error Correction Algorithm: Observation 2

**procedure** correct-PST-M( S,St )
$i := 2$
**while** $i < |S| - 1$ **do**
  **if** $|St_{i-1} - St_i| \geq 2 \wedge |St_{i+1} - St_i| \geq 2$
  $\wedge\ |S_i - S_{i-1}| \leq 7 \wedge |S_i - S_{i+1}| \leq 7$ **then**
    **if** $St_i > (St_{i-1} + St_{i+1})/2$ **then**
      **if** $S_i$ is playable on $St_i - 1$ **then**
        $St_i := St_i - 1$
      **else**
        $St_{i-1} := St_{i-1} + 1$
        $St_{i+1} := St_{i+1} + 1$
      **end if**
    **else**
      $St_i := St_i + 1$
    **end if**
  **end if**
  $i := i + 1$
**end while**
**return** St

### A. Feature Extraction

Our feature is based on the energy contained in a filter bank, as done in many music instrument classification systems [13], [14].

We, however, choose the center frequencies based on a few qualitative observations specific to the bowed strings. First, the high frequency content from approx. 6kHz to 20kHz is a highly discriminating feature for distinguishing between many strings, except for middle register of D and low register of A. Second, middle frequency (approx. 1kHz<approx.6kHz) is required to differentiate the middle register of D and upper register of A. See, for example, Figure 6, which shows the decision boundary between A string and D string for MIDI note number 77. Notice that the greatest component (i.e. the most discriminating component) is the output of the bandpass filter centered around 1.5kHz. On the other hand, the decision boundary for E and A string, for example, has the significant components concentrated toward high frequency (as shown in Figure 7).

The filter bank consists of eight filters with triangular magnitude response, whose center frequencies are spaced equidistantly in log-frequency axis between 1kHz and 15kHz, and the bandwidth constant in the log-frequency axis. We extract the feature using a frame-length of 2048 samples with overlap of 441 samples, at a sampling rate of 44.1kHz.

### B. Model of the Feature

We model the feature as a Gaussian mixture (GMM) governed by F0-dependent parameters, because literatures suggest that instrument features are typically strongly dependent on the fundamental frequency [13], [14]. From

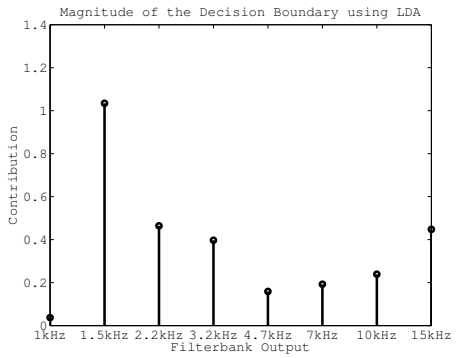


Figure 6. Decision boundary between D and A string for MIDI note number 77.

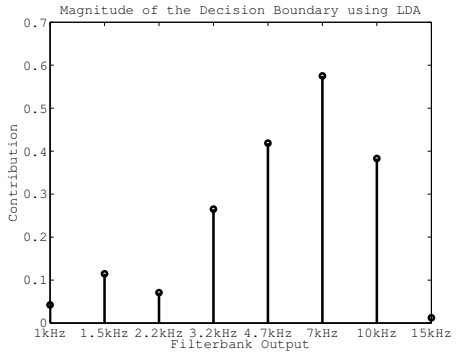


Figure 7. Decision boundary between E and A string for MIDI note number 77.

observation, we noted that the features used in this study are strongly dependent on F0, but within a $\Delta c$-neighborhood, where $\Delta c$=100 cents, only first order effects are observed for the most part.

We model the feature using a zeroth-order approximation within a $\Delta c$-neighborhood of every half note. That is, we divide the frequency axis into $K$=50 regions spaced $\Delta c$ cents apart, such that the $k$th ($K > k \geq 0$) region $R_k$ is described as follows:

$$R_k \in [c_{min} + (k - 0.5)\Delta c, c_{min} + (k + 0.5)\Delta c), \quad (1)$$

where $c_{min}$ is the frequency of the lowest pitch of the violin.

Given an input data $(x, c)$ where $x$ is the feature vector and $c$ is the F0, we assume it came from one of $M$ GMMs ($M = 4$) parametrized by $c$, each of which describes a violin string. First, we associate each data onto one of $K$ cluster as follows:

$$k = \text{round}((c - c_{min})/\Delta c) \quad (2)$$

Next, we determine the cluster-dependent mean of the entire feature set, ave($\mathbf{X_k}$), given a data matrix $\mathbf{X}_k \in \mathbb{R}^{D \times L}$, each of the $L$ rows containing $D$-dimensional feature data that has been assigned to the $k$th cluster:

$$\text{ave}(\mathbf{X_k}) = \mathbf{X}_k \mathbf{1}_L / L \quad (3)$$

where $\mathbf{1_k}$ is a column vector of length k, all the entries of which are 1. We define error $\mathbf{e}$ as the features after subtracting the average.

We then model the likelihood of the error term for the $k$th cluster for the $i$th string, $\mathbf{e}$, as a J-mixture GMM (J=5):

$$p_{\mathbf{e}}(x|i,k) = \sum_{j=1}^{J} \phi_{j,k}(i) \mathcal{N}\left(x|\mu_{j,k}(i), \mathbf{\Sigma}_{j,k}(i)\right) \quad (4)$$

where $\sum_j \phi_{j,k}(i) = 1$, $\phi_{j,k}(i)$, $\mu_{j,k}(i)$ and $\mathbf{\Sigma}_{j,k}(i)$ are $j$th weight, mean vector and covariance matrix, respectively, for the $i$th string in F0 cluster k. $\mathcal{N}(x|u, S)$ is a multivariate normal distribution parametrized by mean $u$ and covariance matrix $S$. We also set the prior on the $i$th string for $k$th F0 cluster, $p_{\text{St}}(i|k)$, as follows:

$$p_{\text{St}}(i|k) = s(k, \text{St}_i) / \sum_{j=1}^{4} s(k, \text{St}_j) \quad (5)$$

Here, $s(k, \text{St}_i)$ is 1 if the string $\text{St}_i$ is playable in $k$th F0 cluster, and 0 otherwise.

Then, for an input $(x, c)$ that has been associated with cluster $k$, we find the maximum a posteriori (MAP) estimate of the bowed string, $\hat{\text{St}}(x, c)$ as follows:

$$\begin{aligned} \hat{\text{St}}(x, c) &= \arg\max_i p(i|x, k) \\ &= \arg\max_i \; p_{\text{St}}(i|k) p_{\mathbf{e}}(x - \text{ave}(\mathbf{X}_k)|i, k) \end{aligned} \quad (6)$$

### C. Bowed String Classification

For each note, we estimate the bowed string by first evaluating the MAP estimate for each frame the note is played. Then, the string that had the greatest count of MAP estimate is chosen as the bowed string. Such voting method was chosen as it offered the greatest accuracy compared to other methods considered. Specifically, we set the estimated string for the $n$th note, $\text{St}_n$ which starts at frame $b_n$ and ends at $e_n$ as the following:

$$\text{St}_n = \arg\max_{m \in \{E,A,D,G\}} \left| \left\{ j \in [b_n, e_n] | \hat{\text{St}}(x^{(j)}, c^{(j)}) = m \right\} \right| \quad (7)$$

where $x^{(j)}$ and $c^{(j)}$ are the feature and the pitch, respectively, for the $j$th frame.

## VI. Adaptation of the Model

Two different violins playing on two different brands of strings typically sound very different – they have significantly different acoustic characteristics. Therefore, we need to adapt the training data to the environment (the instrument and the brand of strings used) of the violin used in a recording from which we need to extract the bowed string sequence.

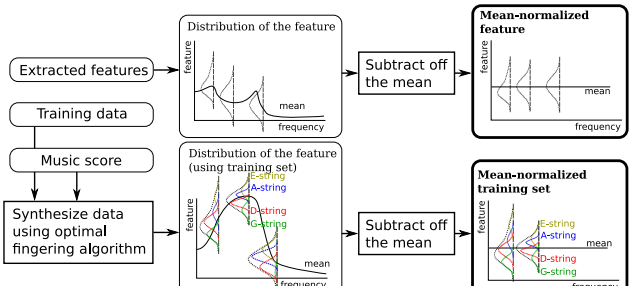


Figure 8. Overview of mean-normalization

We adapt the model by normalizing the mean of the features for each frequency cluster, as shown in Figure 8. Let $X_k$ be the data matrix obtained from the recording for frequency cluster $k$, and let $X_{k,t}$ be the data matrix obtained by synthesizing the music score using the features from the training data. Then, we replace $\mathrm{ave}(X_k)$ defined in equation (3) by $\mathbf{X}_k\mathbf{1}_L/L - \mathbf{X}_{k,t}\mathbf{1}_L/L$. Finally, we evaluate equation (6) to evaluate the posterior.

Our adaptation is based on two assumptions. First, we assume that the main effect of changing the environment is the change in the frequency-dependent mean of the features $\mathrm{ave}(\mathbf{X_k})$. Hence, we assume that once the means are normalized, two recordings using exactly same playing parameters played on different environments would have identical distribution of features. Second, we assume that, given a sufficiently long piece, the distribution of the bowed string for each note is identical to that generated from optimal fingering algorithm. In other words, we assume that the differences of bowed string sequence arising out of personal taste average out to nil, and in a long run, violinists play similar to fingerings generated by the optimal fingering algorithm [2].

## VII. Experiment

We use Yamaha Silent Violin SV150 for training. For validation, we first assess closed-instrument performance using SV150, and open-instrument performance using an acoustic violin (Bisiach 1904). We record the validation data using (a) the same set of strings used for training on SV150, (b) SV150 on *different* set of strings, and (c) acoustic violin on different set of strings. We shall denote (a) as "SS" (as in "same violin, same set of strings"), (b) as "SD" (same, different) and (c) as "DD."

For training data, we first record three two-octave chromatic scales starting at the open string of each note. Next, we record, for each string, a two-octave frequency sweep over duration of five minutes, starting at the open string. These are recorded such that for each note, we span the possible range of sound produced on the note on a particular string. This is accomplished by playing each note gradually from weakest possible sound (*pianissimo possibile*) to strongest possible sound (*fortissimo possibile*) on six different contact points of the bow.

For validation, three excerpts are chosen from standard violin repertoire, as described in Table I. Each excerpt is recorded twice, each with distinctly different fingering, which we denote F.1 and F.2. These were determined by consulting professional violinists. F.1 focuses on choosing the string pattern in concordance with the notion of "voicing." F.2 emphasizes playability for beginners. Roughly speaking, these emphases correspond to F.2 having more string crossings (change of strings) than F.1, and F.1 being more technically challenging than F.2.

Table I
Description of the Pieces Used

| | Name of the Piece | Length | Start | End |
|---|---|---|---|---|
| P1 | Mendelssohn Op.64 (Mvt. 2) | 75 notes | m. 9 | m. 27 |
| P2 | Brahms Op.100 (Mvt. 3) | 41 notes | m. 1 | m. 12 |
| P3 | Brahms Op.100 (Mvt. 3) | 91 notes | m. 1 | m. 15 |

We perform two experiments to answer the following questions. First, assuming that we are able to obtain audio data required for adaptation as described in Section VI, would adaptation offer greater robustness? Second, how effective is each of our error correction algorithms?

### A. Evaluation of Mean Normalization

We record a two-octave chromatic scale with a specific fingering using setups SD and DD. For each note, we vary the dynamics of the note from the softest possible to strongest possible. Next, we manually extract portions of the training data (SS), such that the data represents a two-octave chromatic scale of same duration synthesized from the training data.

We first compute the recognition accuracy of the pieces listed in Table I (414 notes total). Then, the data matrix obtained using two-octave chromatic scales are used to normalize the mean of each of the pieces using adaptation method described in Section VI. Finally, we compute the recognition accuracy of the mean-normalized data.

### B. Evaluation of Error Correction

We use the mean-normalized data from Experiment A to test our anomaly correction algorithm. First, the recognition accuracy by using one of the three algorithms is evaluated. Next, accuracy by using all of the algorithms is evaluated.

To determine how well our algorithms are capable of (a) detecting errors and (b) correcting them properly, we evaluate the precision, recall and $F_{1/2}$-measure. Precision $P$ is defined as the ratio between number of correct fixes to the total number of fixes made, recall $R$ as the ratio between number of correct fixes and the total number of incorrectly identified strings, and $\mathrm{F}_{1/2}$-measure as follows:

$$F_\beta = \left((1+\beta^2)PR\right) / \left(\beta^2 P + R\right)|_{\beta=1/2} \qquad (8)$$

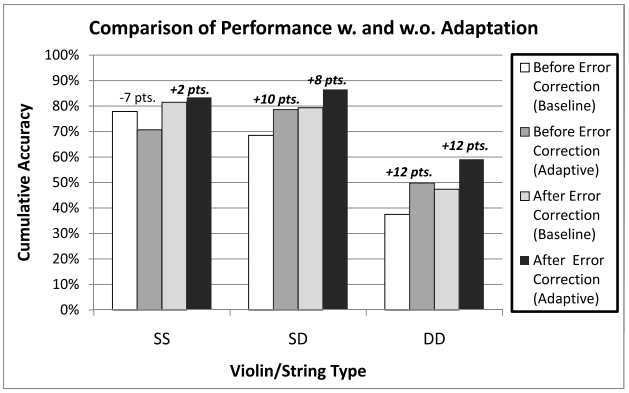


Figure 9. Evaluation of the Adaptation

## VIII. Results

The recognition accuracies as evaluated in Experiment A are shown in Figure 9. We note a significant increase in accuracy for SD (same violin, different strings), and DD (different violin, different strings). Therefore, mean normalization improves the robustness of our method against using different strings or instruments.

The F-measures and the recognition accuracies obtained in Experiment B are shown in Table II. Cumulative accuracy (the sum of correctly guessed notes for all data divided by the total number of notes) is denoted as "cumul." The algorithm that caused the greatest increase in recognition accuracy is shown in boldface. We note that the cumulative accuracy increases by 13 points for SS, 8pt for SD, and 9pt for DD. The only instance of decreasing accuracy is observed in P2F1 of DD.

## IX. Discussion

From Experiment A, we note that our adaptation and error correction resulted in a significant improvement in cumulative recognition accuracy, especially showing robustness against variation of strings (setup SD). On the other hand, setup DD shows insufficient result, meaning that our method is not robust against variation of instruments. The fact that adaptation results in similar performance improvement for both SD and DD suggests a need for better audio classifier.

From Experiment B, we also note that P2F1 (DD) in Table II has an anomalously high accuracy compared to the accuracy of other pieces from "DD." This is possibly because P2F1 is played in the lower register of the instrument. Therefore, there are fewer strings that are physically playable, which in turn reduced the number of classes to consider. P1 and P3 involve higher registers compared to P2, which may explain the tendency for P2 to have high accuracy in general.

We note that imposing rule MORD caused only a nominal improvement in accuracy compared to the two error correction algorithms. This is probably because there are only few mordents in our dataset (twelve total). Since mordents occur infrequently in general, we expect little improvement by imposing MORD for other pieces as well.

Sometimes, the "errors" are correctly identified but wrong corrections are made, as suggested by low precision in DD, Table II. This happens because there may be more than one way to fix an error. For example, Figure 10 has an error that may be corrected by imposing rule PST-D, but there are two ways to fix it.

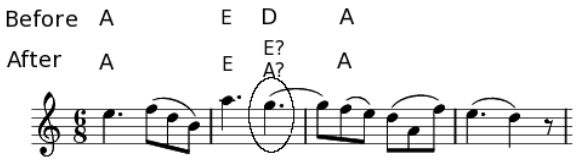


Figure 10. Example of an Ambiguous Error

Though unmentioned, we originally incorporated higher order terms (1st, 1st+2nd) to model the F0-dependent average of our features. That is, we assumed that in a $\Delta$c cent neighborhood of $c_0$, we could approximate the feature $(c, x)$ as follows:

$$x \approx x_0 + (c - c_0)\frac{\delta x}{\delta c} + (c - c_0)^2\frac{\delta^2 x}{\delta c^2} + \text{Error} \qquad (9)$$

where $\delta x/\delta c$ and $\delta^2 x/\delta c^2$ are regression coefficients determined through polynomial regression. However, this significantly lowered the accuracy, especially for setup SD and DD. We believe this occurs because using higher-order terms overfit the feature to a specific instrument configuration.

## X. Conclusion and Future Work

This paper presented a method for estimating the sequence of bowed strings on a violin by audio-based classification and three error correction algorithms by analyzing the musical context. We also presented a way of adapting trained data to a new instrument setup by normalizing the mean of the feature. We have shown that mean normalization improves the robustness of our method against variation of strings used, and have also shown that context-dependent error correction improves the recognition accuracy.

We seek to improve on our method's robustness against variation of violins, as result "DD" is still insufficiently low for our application. This might be achieved through better feature selection and adaptation scheme. Also, we seek to apply our method for fingering estimation by combining our research with optimal fingering estimation algorithm.

## Acknowledgment

This work is supported by Grant-in-aid for Scientific Research (S) and CREST-MUSE of JST. We would like to thank violinists P. Klinger, Dr. P. Sunwoo and Dr. J. Choi for stimulating and inspiring discussions on violin fingering.

Table II
RESULT OF EXPERIMENT 2

| Piece (Violin/String) | Precision | Recall | $F_{1/2}$ | Before | PST-M | PST-D | MORD | All | Increase |
|---|---|---|---|---|---|---|---|---|---|
| P1F1 (SS) | 0.71 | 0.44 | 0.63 | 71% | 72% | **78%** | 77% | 84% | 13pt |
| P1F2 (SS) | 0.67 | 0.37 | 0.57 | 71% | 73% | **82%** | 62% | 82% | 11pt |
| P2F1 (SS) | n/a | n/a | n/a | 95% | 95% | 95% | 95% | 95% | 0pt |
| P2F2 (SS) | n/a | n/a | n/a | 98% | 98% | 98% | 98% | 98% | 0pt |
| P3F1 (SS) | 0.81 | 0.46 | 0.71 | 63% | 63% | **76%** | 65% | 80% | 17pt |
| P3F2 (SS) | 0.90 | 0.49 | 0.77 | 51% | 48% | **73%** | 52% | 75% | 24pt |
| Cumul. (SS) | | | | 71% | 71% | **81%** | 71% | 83% | 13pt |
| P1F1 (SD) | 0.00 | 0.00 | n/a | 79% | **84%** | 79% | 82% | 79% | 0pt |
| P1F2 (SD) | 1.00 | 0.84 | 0.96 | 86% | 89% | **97%** | 87% | 98% | 12pt |
| P2F1 (SD) | n/a | 0.00 | n/a | 90% | 90% | 90% | 90% | 90% | 0pt |
| P2F2 (SD) | n/a | 0.00 | n/a | 93% | 93% | 93% | 93% | 93% | 0pt |
| P3F1 (SD) | 0.67 | 0.30 | 0.53 | 64% | 67% | **73%** | 65% | 75% | 11pt |
| P3F2 (SD) | 1.00 | 0.61 | 0.89 | 69% | 69% | **87%** | 71% | 88% | 19pt |
| Cumul. (SD) | | | | 79% | 81% | **86%** | 80% | 87% | 8pt |
| P1F1 (DD) | 0.42 | 0.20 | 0.34 | 45% | 45% | **50%** | 45% | 55% | 11pt |
| P1F2 (DD) | 0.58 | 0.36 | 0.51 | 54% | 57% | **67%** | 57% | 71% | 16pt |
| P2F1 (DD) | -1.00 | -0.33 | -0.71 | 85% | 85% | 80% | 85% | 80% | -5pt |
| P2F2 (DD) | 0.00 | 0.00 | n/a | 56% | 56% | 56% | 56% | 56% | 0pt |
| P3F1 (DD) | 0.15 | 0.04 | 0.10 | 39% | 40% | 40% | 40% | 41% | 3pt |
| P3F2 (DD) | 0.74 | 0.30 | 0.57 | 39% | 37% | **55%** | 40% | 57% | 19pt |
| Cumul. (DD) | | | | 50% | 50% | **56%** | 51% | 59% | 9pt |

## REFERENCES


[1] H. Lu, B. Zhang, Y. Wang, and W. K. Leow, "iDVT: an interactive digital violin tutoring system based on audio-visual fusion," in *MM'08: Proc. of the 16th ACM int'l conf. on Multimedia*. New York, NY, USA: ACM, 2008, pp. 1005–1006.

[2] S. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[3] C. Kerdvibulvech and H. Saito, "Vision-based detection of guitar players? fingertips without markers," *Int'l Conf. on Computer Graphics, Imaging and Visualization*, vol. 0, pp. 419–428, 2007.

[4] Y. Yonebayashi, H. Kameoka, and S. Sagayama, "Automatic decision of piano fingering based on a hidden markov models," in *IJCAI*, 2007, pp. 2915–2921.

[5] C.-C. Lin and D. S.-M. Liu, "An intelligent virtual piano tutor," in *VRCIA '06: Proceedings of the 2006 ACM int'l conf. on Virtual reality continuum and its applications*. New York, NY, USA: ACM, 2006, pp. 353–356.

[6] D. P. Radicioni *et al.*, "A Segmentation-Based Prototype to Compute String Instruments Fingering," in *CIM'04*, R. Parncutt, A. Kessler, and F. Zimmer, Eds., Graz, Austria, 2004.

[7] I. M. Yampolsky, *Principles of Violin Fingering*. Oxford University Press, 1967.

[8] C. Flesch, *The Art of Violin Playing*, second edition ed. New York, NY, USA: Carl Fischer, L.L.C., 2000, vol. Book One.

[9] R. Stowell, *The Cambridge Companion to the Violin*. Cambridge: Cambridge University Press, 1992.

[10] A. Krishnaswamy and J. Smith, "Inferring control inputs to an acoustic violin from audio spectra," *ICME'03*, vol. 1, pp. 733–736, 2003.

[11] N. Hu *et al.*, "Polyphonic audio matching and alignment for music retrieval," in *WASPAA 2003*, Oct. 2003, pp. 185–188.

[12] K. Itoyama *et al.*, "Integration and adaptation of harmonic and inharmonic models for separating polyphonic musical signals," in *ICASSP'07*, vol. 1, April 2007, pp. I–57–I–60.

[13] P. Herrera-Boyer, "Automatic classification of musical instrument sounds," *J. of New Music Research*, vol. 32, no. 1, pp. 3–21(19), March 2003.

[14] T. Kitahara *et al.*, "Pitch-dependent identification of musical instrument sounds," *App. Intelligence*, vol. 23, no. 3, pp. 267–275, 2003.